Marginal Pseudo-Likelihood Learning of Discrete Markov Network Structures

Pensar, Johan

2017-12

Pensar , J , Nyman , H , Niiranen , J & Corander , J 2017 , ' Marginal Pseudo-Likelihood Learning of Discrete Markov Network Structures ' , Bayesian analysis , vol. 12 , no. 4 , pp. 1195-1215 . https://doi.org/10.1214/16-BA1032

http://hdl.handle.net/10138/231261
https://doi.org/10.1214/16-BA1032

unspecified
publishedVersion

Bayesian Analysis (2017) **12**, Number 4, pp. 1195–1215

Marginal Pseudo-Likelihood Learning of Discrete Markov Network Structures

Johan Pensar[*], Henrik Nyman[†], Juha Niiranen[‡], and Jukka Corander[§,¶]

Abstract. Markov networks are a popular tool for modeling multivariate distributions over a set of discrete variables. The core of the Markov network representation is an undirected graph which elegantly captures the dependence structure over the variables. Traditionally, the Bayesian approach of learning the graph structure from data has been done under the assumption of chordality since non-chordal graphs are difficult to evaluate for likelihood-based scores. Recently, there has been a surge of interest towards the use of regularized pseudo-likelihood methods as such approaches can avoid the assumption of chordality. Many of the currently available methods necessitate the use of a tuning parameter to adapt the level of regularization for a particular dataset. Here we introduce the marginal pseudo-likelihood which has a built-in regularization through marginalization over the graph-specific nuisance parameters. We prove consistency of the resulting graph estimator via comparison with the pseudo-Bayesian information criterion. To identify high-scoring graph structures in a high-dimensional setting we design a two-step algorithm that exploits the decomposable structure of the score. Using synthetic and existing benchmark networks, the marginal pseudo-likelihood method is shown to perform favorably against recent popular structure learning methods.

Keywords: Markov networks, structure learning, pseudo-likelihood, non-chordal graph, Bayesian inference, regularization.

1 Introduction

Markov networks, also known as undirected graphical models, represent a ubiquitous modeling framework for multivariate distributions, with applications covering areas such as statistical physics, computer vision, computational biology and sociology (see Lauritzen, 1996; Koller and Friedman, 2009). In this paper, we consider discrete Markov networks, adopting the common assumption of positive distributions. To enable modeling of high-dimensional distributions, Markov networks exploit structure in the distribution in form of assumptions of conditional independence among the involved variables. Markov networks use an undirected graph to compactly capture and represent the dependence structure over variables. Although the ultimate goal of a Markov network is to efficiently represent a multivariate distribution, the graph alone is also useful for gaining insight into complex dependency patterns among large collections of variables.

[*]Dept. of Mathematics and Statistics, Åbo Akademi University, Finland, johan.pensar@abo.fi
[†]Dept. of Mathematics and Statistics, Åbo Akademi University, Finland, henrik.nyman@abo.fi
[‡]Dept. of Mathematics and Statistics, University of Helsinki, Finland, juha.niiranen@helsinki.fi
[§]Dept. of Mathematics and Statistics, University of Helsinki, Finland, jukka.corander@helsinki.fi
[¶]Dept. of Biostatistics, University of Oslo, Norway

 DOI: 10.1214/16-BA1032

In this paper, we consider the problem of learning the graph structure from a collection of independent and identically distributed samples drawn from the distribution of a Markov network. This task is very challenging due to the extremely vast model space, for d variables the number of possible graphs is $2^{d(d-1)/2}$. Moreover, one needs a sound approach for efficiently comparing the plausibility of different graphs for a given dataset. For the related class of Bayesian networks, the Bayesian score (Heckerman et al., 1995) has become the most popular measure for this purpose. In addition to the solid performance, an important reason behind the popularity of the Bayesian score is that the marginal likelihood can be efficiently evaluated by a closed-form expression. Unfortunately, likelihood-based techniques are in general intractable for non-chordal Markov networks due to a normalizing factor known as the partition function. By restricting the graph space to chordal graphs, it is possible to perform likelihood-based inference since the distribution of chordal Markov networks can be genuinely factorized according to the graph. However, since the chordality assumption is restrictive and may seriously bias learning of dependencies among variables, considerable interest has been targeted towards making learning of non-chordal networks tractable in high-dimensional settings.

As a computationally more convenient alternative to the likelihood function, Besag (1975) introduced the pseudo-likelihood function which approximates the likelihood function by a product of local conditional likelihoods of the random variables involved in the modeled system. The pseudo-likelihood approach has during the last few decades paved the way for an array of learning methods applicable on larger systems (Ji and Seymour, 1996; Csiszár and Talata, 2006; Meinshausen and Bühlmann, 2006; Höfling and Tibshirani, 2009; Ravikumar et al., 2010; Aurell and Ekeberg, 2012; Ekeberg et al., 2013; Lowd and Davis, 2014; Barber and Drton, 2015). In particular, Meinshausen and Bühlmann (2006) introduced the popular regression-based Lasso for Gaussian graphical models which was later adapted to discrete Ising models (Ravikumar et al., 2010; Barber and Drton, 2015). In addition to the mentioned learning methods, Heckerman et al. (2000) introduced a pseudo-likelihood-type model class known as dependency networks. Again, the main motivation behind the proposed model class was the possibility of performing efficient model learning.

The main contribution of this work is introducing the marginal pseudo-likelihood (MPL) as a tractable alternative to the marginal likelihood in the context of learning the graph structure of a Markov network. We show that the MPL has a built-in regularization of the model complexity as a result of marginalization over some graph-specific nuisance parameters associated with the pseudo-likelihood. In particular we show that the resulting score function can be considered a small sample analytical version of the consistent pseudo-Bayesian information criterion (PIC) by Csiszár and Talata (2006). The MPL is well-suited for high-dimensional applications since it can be evaluated in closed form for both chordal and non-chordal Markov networks. Moreover, we show that the factorization of the MPL makes it particularly convenient for search algorithms based on single edge changes.

The structure of the remaining article is as follows. In Section 2, the basic properties of Markov networks are reviewed and the structure learning problem is presented. In Section 3, we derive the marginal pseudo-likelihood score, examine its asymptotic properties, and discuss related work. In Section 4 we examine the computational complexity

of the MPL and explain how the factorization of the MPL can be used to speed up learning procedures. In addition, we introduce a search algorithm which can be used to find high-scoring graphs in a high-dimensional setting. Section 5 demonstrates the favorable performance of our method against other popular recent alternatives in extensive numerical experiments, and finally Section 6 provides some additional remarks and conclusions. The supplementary Appendix (Pensar et al., 2016) contains a proof of the consistency theorem, pseudocode of the search algorithms, and detailed results from the experiments.

2 Structure learning of Markov networks

We consider a set of d discrete random variables $X = \{X_1, \ldots, X_d\}$ where each variable X_j takes values from a finite set of outcomes $\mathcal{X}_j$. By letting $V = \{1, \ldots, d\}$ denote the indices of the variables, a subset $S \subseteq V$ of the variables is denoted by $X_S = \{X_j\}_{j \in S}$. The corresponding joint outcome space is specified by the Cartesian product $\mathcal{X}_S = \times_{j \in S} \mathcal{X}_j$. The cardinality of an outcome space is denoted by $|\mathcal{X}_S|$. We use a lowercase letter x_S to denote that the variables have been assigned a specific joint outcome in $\mathcal{X}_S$. Finally, we let $\mathbf{x}$ denote a dataset consisting of n i.i.d. complete joint observations over the d variables.

2.1 Markov networks

A Markov network over X is an undirected probabilistic graphical model that compactly represents a joint distribution over the variables. The dependence structure over the d variables is specified by an undirected graph $G = (V, E)$ where the nodes $V = \{1, \ldots, d\}$ correspond to the indices of the variables and the edges $E \subseteq \{V \times V\}$ represent dependencies among the variables. A node i is a neighbor of j (and vice versa) if $\{i, j\} \in E$. The set of all neighbors of a node j is called the Markov blanket of node j and is denoted by $mb(j)$. A clique in a graph is a subset of nodes, $C \subseteq V$, for which every pair of nodes are connected by an edge, that is $\{i, j\} \in E$ if $i, j \in C$. A clique C is said to be maximal if for any superset of nodes $C' \supset C$, C' is not a clique. The set of maximal cliques associated with a graph is denoted by $\mathcal{C}(G)$. The complete set of undirected graphs is denoted by $\mathcal{G}$. As is common in the graphical model literature, the terms node and variable are occasionally used interchangeably in this article.

The absence of edges in the graph $G = (V, E)$ of a Markov network encodes statements of conditional independence as characterized by the global Markov property. More specifically, for any three disjoint subsets A, B, S of V, the variables X_A and X_B are conditionally independent given X_S if S separates A and B. To fully specify a Markov network, one must in addition to the graph also define a probability distribution that satisfies the conditional independence restrictions imposed by the graph G. A distribution is said to be faithful to G if it does not satisfy any additional independencies not conveyed by the graph. We assume that the distribution is positive, meaning that $p(x) > 0$ for all $x \in \mathcal{X}$. The positivity assumption ensures that the joint distribution of

a Markov network can be factorized over the maximal cliques in the graph according to

$$p(x) = \frac{1}{Z} \prod_{C \in \mathcal{C}(G)} \phi(x_C), \tag{1}$$

where $\phi(x_C) : \mathcal{X}_C \to \mathbb{R}_+$ is a clique factor (or potential) and $Z = \sum_{x \in \mathcal{X}} \prod_{C \in \mathcal{C}(G)} \phi(x_C)$ is a normalizing constant known as the partition function. Alternatively, Markov networks are often parameterized in terms of a log-linear model (see Whittaker, 1990).

2.2 Structure learning

There are two main tasks associated with fitting graphical models to data; parameter estimation and structure learning. In this work, we focus entirely on the latter. By structure learning, we refer to the process of inferring the dependence structure from a set of data assumed to be generated from an unknown Markov network. In many applications, the structure is a goal in itself in the sense that one wants merely to gain a qualitative insight into the dependence structure of an underlying process. If the distribution needs also to be explicitly estimated, this can be achieved by using any of several existing parameter estimation methods conditional on the fixed structure learned by our approach (see e.g. Liu and Ihler, 2012; Mizrahi et al., 2014).

The most fine-grained structure learning methods aim at recovering distinct features in the log-linear parameterization, this approach is commonly referred to as feature selection (Pietra et al., 1997; Lee et al., 2006; Höfling and Tibshirani, 2009; Ravikumar et al., 2010; Lowd and Davis, 2014). In contrast to the very specific feature selection problem, the model space of our approach is formulated in terms of the graph structure alone. Although a very detailed structure may better emulate the properties of a distribution without imposing redundant parameters, a drawback is the risk of overfitting the structure through long specialized features. Since every pair of variables in a feature results in an edge, sparsity in the number of features does not in general correspond to sparsity in the number of edges in the graph (see Koller and Friedman, 2009). This can pose problems for example from the perspective of performing inference in the model, since inference algorithms are often designed to exploit sparsity in the graph. Additionally, in terms of knowledge discovery, a dense graph may in the worst case hide the primary layer of the dependency pattern.

Structure learning methods can roughly be divided into two categories; constraint-based and score-based. Constraint-based approaches aim at inferring the structure through a series of independence tests (Spirtes et al., 2000; Tsamardinos et al., 2003; Bromberg et al., 2009; Anandkumar et al., 2012). The score-based approach formulates structure learning as an optimization problem. This requires a score function by which the plausibility of the different candidates can be evaluated. Additionally, this requires an optimization algorithm for finding high-scoring graphs since an exhaustive evaluation is in general infeasible. The local approach of constraint-based methods allows them to scale up well but it makes them more sensitive to failures in the individual tests. Score-based methods work on a global level by considering the whole structure at once making them less sensitive to local failures, however, it has a negative effect on

their scalability. Although the MPL as such falls in the score-based category, under the introduced search algorithm, our final method could rather be considered a hybrid by which we aim to achieve scalability as well as reliable performance.

3 Marginal pseudo-likelihood

The most common criterion for evaluating the model fit with respect to a dataset $\mathbf{x}$ is by maximizing the likelihood function $p(\mathbf{x} \mid \theta_G)$ over the model parameters θ_G which specify the distribution for a given graph G. Since the likelihood function attains its maximum value under the complete graph, the expressiveness of the models must be constrained (Chow and Liu, 1968) or regulated by adding a sparsity-promoting penalty function (Akaike, 1974; Schwarz, 1978; Lee et al., 2006).

A well-established alternative to explicitly penalizing the complexity of a model is given by the Bayesian framework. In the Bayesian approach, a graph is scored by its posterior probability given the data,

$$p(G \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid G) \cdot p(G)}{p(\mathbf{x})}.$$

In practice it suffices to consider the unnormalized posterior probability

$$p(G, \mathbf{x}) = p(\mathbf{x} \mid G) \cdot p(G), \tag{2}$$

since $p(\mathbf{x})$ is a normalizing constant that can be ignored when comparing graphs. The key factor of (2) is $p(\mathbf{x} \mid G)$ which is the marginal likelihood of the data given the network structure (also called the evidence). To evaluate the marginal likelihood, one must integrate the likelihood function over all parameter values satisfying the restrictions imposed by the graph according to

$$p(\mathbf{x} \mid G) = \int p(\mathbf{x} \mid \theta_G) \cdot f(\theta_G) d\theta_G,$$

where $f(\theta_G)$ is a prior distribution that assigns a weight to each possible instantiation of θ_G. Since the marginal likelihood accounts for the parameter uncertainty through the prior, it implicitly regulates the fit to the data against the complexity of the network.

A significant drawback of likelihood-based scores is that they, due to the partition function, are extremely hard to evaluate for non-chordal Markov networks. In order to avoid this problem, one can preferably use alternative objective functions that possess favorable properties from a computational perspective. In particular, our score is based on the pseudo-likelihood function by Besag (1975).

3.1 Derivation

The pseudo-likelihood function approximates the likelihood function by a product of conditional likelihood functions according to

$$\hat{p}(\mathbf{x} \mid \theta) = \prod_{j=1}^{d} p(\mathbf{x}_j \mid \mathbf{x}_{V \setminus j}, \theta).$$

For a fixed graph structure G, each variable in a Markov network is conditionally independent of the remaining variables given its Markov blanket. Consequently, the pseudo-likelihood for a fixed graph is given by

$$\hat{p}(\mathbf{x} \mid \theta_G) = \prod_{j=1}^{d} p(\mathbf{x}_j \mid \mathbf{x}_{mb(j)}, \theta_G). \tag{3}$$

In terms of maximization, the pseudo-likelihood approximation offers huge computational savings compared to the true likelihood since the global normalizing constant in the likelihood function is replaced by d local normalizing constants (see e.g. Koller and Friedman, 2009, Section 20.6.1). By replacing the likelihood with the pseudo-likelihood, methods originally based on the maximum likelihood (Schwarz, 1978; Lee et al., 2006) have been extended to work on larger systems (Ji and Seymour, 1996; Csiszár and Talata, 2006; Höfling and Tibshirani, 2009; Ravikumar et al., 2010). Moreover, different pseudo-likelihood-based structure estimators have been shown to be consistent under the assumption that the data is generated by a distribution in the model class (Ji and Seymour, 1996; Csiszár and Talata, 2006; Ravikumar et al., 2010).

From a Bayesian perspective, the structural form of (3) offers an interesting possibility. In fact, under certain simplifying assumptions it enables an analytical evaluation of the integral

$$\hat{p}(\mathbf{x} \mid G) = \int \hat{p}(\mathbf{x} \mid \theta_G) \cdot f(\theta_G) d\theta_G, \tag{4}$$

which is here referred to as the marginal pseudo-likelihood (MPL).

We parameterize the conditional probabilities associated with the pseudo-likelihood function of a graph by

$$\theta_{ijl} = p(X_j = x_j^{(i)} \mid X_{mb(j)} = x_{mb(j)}^{(l)}). \tag{5}$$

The indices $i = 1, \ldots, r_j$ and $l = 1, \ldots, q_j$, where $r_j = |\mathcal{X}_j|$ and $q_j = |\mathcal{X}_{mb(j)}| = \prod_{i \in mb(j)} r_i$, represent the configurations of the variable and its respective Markov blanket. The above set of graph-specific parameters is by no means a compact representation of a Markov network, however, rather than actual model parameters, they should be considered temporary nuisance parameters. Similarly as above, we define counts n_{ijl} representing the number of times in the data $\mathbf{x}$ that the variable X_j has taken on value i given that the Markov blanket $X_{mb(j)}$ has taken on configuration l. The pseudo-likelihood function can now be expressed in terms of our introduced notation by

$$\hat{p}(\mathbf{x} \mid \theta_G) = \prod_{j=1}^{d} \prod_{l=1}^{q_j} \prod_{i=1}^{r_j} \theta_{ijl}^{n_{ijl}}.$$

Under the current parameterization it is easy to make out certain structural similarities between the above pseudo-likelihood function and the likelihood function of a Bayesian network under a standard conditional parameterization (Heckerman et al., 1995, Equation (18)). In a Bayesian network the l-index would be associated with configurations of parent sets instead of configurations of Markov blankets. To carry on the analogy,

the parent sets in Bayesian network must satisfy the acyclicity constraint whereas the Markov blankets must be mutually consistent.

Under certain assumptions listed by Heckerman et al. (1995) the marginal likelihood of a Bayesian network has a nice analytic expression that factorizes variable-wise making it attractive for the task of structure learning. We consider the parameters defined in (5) in terms of the sets

$$\theta_{jl} = \cup_{i=1}^{r_j}\{\theta_{ijl}\},\ \theta_j = \cup_{l=1}^{q_j}\{\theta_{jl}\}, \text{ and } \theta_G = \cup_{j=1}^{d}\{\theta_j\}.$$

One of the fundamental assumptions behind the derivation of the marginal likelihood for Bayesian networks is an assumption regarding global and local parameter independence (Heckerman et al., 1995, Assumption 2) which ultimately justifies a factorization of the parameter prior in a similar fashion as the likelihood. Analogously, we need to factorize the parameter prior in (4) in a similar fashion as the pseudo-likelihood according to

$$f(\theta_G) = \prod_{j=1}^{d} f(\theta_j) = \prod_{j=1}^{d}\prod_{l=1}^{q_j} f(\theta_{jl}), \tag{6}$$

implying that $\theta_j \perp \theta_{j'}$ for $j \neq j'$ and $\theta_{jl} \perp \theta_{jl'}$ for $l \neq l'$. We note that the above assumption violates the properties of a Markov network in the sense that the conditional distributions, represented by our parameters, must satisfy certain algebraic relations for them to be consistent with a distribution of a Markov network (cf. consistent dependency networks, Heckerman et al., 2000). The main justification behind the assumption is computational convenience and it is explicitly or implicitly assumed by most pseudo-likelihood-based techniques.

Under the parameter independence assumption, the integral in (4) can be reordered into a product of local integrals:

$$\hat{p}(\mathbf{x} \mid G) = \int \hat{p}(\mathbf{x} \mid \theta_G) \cdot f(\theta_G) d\theta_G = \prod_{j=1}^{d}\prod_{l=1}^{q_j}\int \prod_{i=1}^{r_j} \theta_{ijl}^{n_{ijl}} \cdot f(\theta_{jl}) d\theta_{jl}.$$

By finally assuming that each parameter set follows a Dirichlet distribution,

$$\theta_{jl} \sim \text{Dirichlet}(\alpha_{1jl}, \ldots, \alpha_{r_j jl}),$$

the MPL is easily solved by standard Bayesian calculations giving the closed-form expression

$$\hat{p}(\mathbf{x} \mid G) = \prod_{j=1}^{d}\prod_{l=1}^{q_j} \frac{\Gamma(\alpha_{jl})}{\Gamma(n_{jl} + \alpha_{jl})} \prod_{i=1}^{r_j} \frac{\Gamma(n_{ijl} + \alpha_{ijl})}{\Gamma(\alpha_{ijl})},$$

where $n_{jl} = \sum_{i=1}^{r_j} n_{ijl}$ and $\alpha_{jl} = \sum_{i=1}^{r_j} \alpha_{ijl}$. We specify the hyperparameters by setting $\alpha_{ijl} = 1/2$ which results in Jeffreys' prior for the multinomial distribution. Note that the MPL under the current assumptions factorizes into variable-wise marginal conditional likelihoods or, since we for computational reasons use the log-version, a sum of variable-wise scores:

$$\log \hat{p}(\mathbf{x} \mid G) = \sum_{j=1}^{d} \log p(\mathbf{x}_j \mid \mathbf{x}_{mb(j)}). \tag{7}$$

We refer to the score of a single variable as the local MPL of that specific variable.

In addition to the MPL, the final score also contains a graph prior $p(G)$ through which it is possible to incorporate any prior beliefs regarding, for example, degree of sparsity. Ideally, the prior should be defined in terms of mutually independent prior beliefs on the individual Markov blankets. This way the convenient decomposition of the MPL (7) is maintained in the final score function and can be exploited to speed up the search procedure. Since our main interest lies in the high-dimensional setting, we take inspiration from the extended Bayesian information criterion (see e.g. Barber and Drton, 2015) and define our graph prior in terms of the number of edges $|E|$ according to

$$\log p(G) = |E| \log(d) - C,$$

where C is a normalizing constant that can be ignored when comparing graphs. The prior further penalizes the inclusion of an edge by a constant term determined by the number of nodes d. The effect of the prior will be strongest in situations when the number of nodes d is large compared to the sample size n. On the other hand, the effect of the prior vanishes as the sample size increases. Note that the proposed prior can be reformulated in terms of the size of the individual Markov blankets $|mb(j)|$ according to

$$\log p(G) = \sum_{j=1}^{d} \log p(mb(j)) = \sum_{j=1}^{d} \big[\frac{|mb(j)|}{2} \log(d)\big] - C,$$

such that the decomposition of the final score function is maintained.

3.2 Asymptotic behavior

Consistency is an important asymptotic property preferably satisfied by a scoring function. By consistency we mean that, under the assumption that the generating distribution is faithful to a Markov network structure, the score will favor the true graph when the sample size tends to infinity. The following theorem establishes that the MPL-based graph estimator is indeed consistent.

Theorem 1. *Let $G^* \in \mathcal{G}$ be the true graph structure of a Markov network, over d variables, which has a positive and faithful distribution. Let $\mathbf{x}$ be a sample of size n generated from the model. The local MPL estimator*

$$\widehat{mb}(j) = \underset{mb(j) \subseteq V \setminus j}{\arg\max}\ p(\mathbf{x}_j \mid \mathbf{x}_{mb(j)})$$

is consistent in the sense that $\widehat{mb}(j) = mb^*(j)$ eventually almost surely as $n \to \infty$ for $j = 1, \ldots, d$. Consequently, the global MPL estimator

$$\hat{G} = \underset{G \in \mathcal{G}}{\arg\max}\ \hat{p}(\mathbf{x} \mid G)$$

is consistent in the sense that $\hat{G} = G^*$ eventually almost surely as $n \to \infty$.

Proof. See Section S1 in the Appendix.

Consistency is a reassuring property validating the concept of MPL theoretically, however, it is also important to investigate how well a score function performs in practice for limited sample sizes. In Section 5 we do a series of large-scale numerical simulations to investigate how well our MPL method performs in comparison with other methods.

3.3 Related work

The concept of using pseudo-likelihood for Markov network learning is closely related to a class of models known as dependency networks (Heckerman et al., 2000). Similar to the pseudo-likelihood, the distribution of a dependency network is represented by variable-wise conditional distributions. The main advantage of dependency networks is that they are very convenient from a learning perspective, which is also the main motivation behind the pseudo-likelihood. However, in contrast to a Markov network, a dependency network is in general not consistent with a single joint distribution, that is, there is no joint distribution from which the conditional distributions can be obtained through inference.

Regression-based techniques (Meinshausen and Bühlmann, 2006; Ravikumar et al., 2010; Lowd and Davis, 2014; Barber and Drton, 2015) can be viewed as learning the structure of a general dependency network, which is then transformed into a Markov network structure. Analogously, our method can be viewed as finding a bi-directional dependency network structure, which is then converted into an undirected graph with the same structural adjacencies. In fact, the MPL of a Markov network is essentially equivalent to what would be the marginal likelihood of a (non-consistent) bi-directional dependency network under a complete parameterization. Bi-directionality corresponds to structural consistency in that $i \in mb(j)$ iff $j \in mb(i)$. Note that bi-directionality is a prerequisite for a dependency network to be consistent with the joint distribution of a Markov network (Heckerman et al., 2000, Theorem 4).

4 Optimizing the marginal pseudo-likelihood

4.1 Computational complexity

We begin this section by examining the computational complexity of the MPL. The (log-)MPL is calculated by the sum

$$\sum_{j=1}^{d}\Big[\sum_{l=1}^{q_j}\big[\log\Gamma(\alpha_{jl})-\log\Gamma(n_{jl}+\alpha_{jl})+\sum_{i=1}^{r_j}\left[\log\Gamma(n_{ijl}+\alpha_{ijl})-\log\Gamma(\alpha_{ijl})\right]\big]\Big],$$

which consists of $\sum_{j=1}^{d} q_j(2+2r_j)$ terms. Since $r_j = |\mathcal{X}_j|$ does not depend on the graph, the number of terms, associated with a node j, is mainly determined by the number of Markov blanket configurations, q_j, which grows exponentially with the size of the Markov blanket. Still, it is important to note that the partial sum associated with a

specific combination of j and l does not contribute to the MPL if $n_{jl} = 0$, that is, if the corresponding Markov blanket configuration is not represented in the data. Consequently, the maximum number of terms evaluated by a non-naive implementation is $\sum_{j=1}^{d} \min(q_j, n)(2 + 2r_j)$ where n is the number of observations in the dataset. Furthermore, for a large Markov blanket, the number of distinct configurations present in a dataset is, in practice, usually far less than $\min(q_j, n)$.

If we look at the MPL from an optimization perspective, it is easy to see that its variable-wise decomposition (7) makes it a convenient candidate for search algorithms based on local changes. To compare the plausibility of two graphs, $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$, we can calculate the log-ratio of their MPLs,

$$\log K(G_1, G_2) = \log \hat{p}(\mathbf{x} \mid G_1) - \log \hat{p}(\mathbf{x} \mid G_2), \tag{8}$$

which is basically a pseudo-version of log-Bayes factor. Assume there is a single edge difference,

$$\{E_1 \cup E_2\} \setminus \{E_1 \cap E_2\} = \{i, j\},$$

between the graphs. This implies that $mb(i)$ and $mb(j)$ are the only Markov blankets that differ in the two graphs. Consequently, log-Bayes pseudo-factor is simply evaluated by

$$\begin{aligned} \log K(G_1, G_2) = &\log p(\mathbf{x}_i \mid \mathbf{x}_{mb_1(i)}) + \log p(\mathbf{x}_j \mid \mathbf{x}_{mb_1(j)}) \\ &- \log p(\mathbf{x}_i \mid \mathbf{x}_{mb_2(i)}) - \log p(\mathbf{x}_j \mid \mathbf{x}_{mb_2(j)}) \end{aligned} \tag{9}$$

since the rest of the terms cancel each other out.

4.2 A search algorithm

The straightforward MPL-based optimization problem is formulated by

$$\arg\max_{G \in \mathcal{G}} \big[\log \hat{p}(\mathbf{x} \mid G) + \log p(G) \big],$$

or given that our prior factorizes,

$$\begin{aligned} &\arg\max_{\{mb(j) \subseteq V \setminus j\}_{j \in V}} \sum_{j=1}^{d} \big[\log p(\mathbf{x}_j \mid \mathbf{x}_{mb(j)}) + \log p(mb(j)) \big] \\ &\text{subject to} \quad i \in mb(j) \Rightarrow j \in mb(i) \text{ for all } i, j \in V. \end{aligned} \tag{10}$$

We refer to the above problem as global MPL optimization.

Due to the vast discrete optimization space, the above problem is clearly intractable for large systems. Hence, we need to construct an algorithm for finding approximate solutions of satisfactory quality in a reasonable time. To ensure applicability in a genuinely high-dimensional setting, the algorithm is designed to exploit the structural decomposition of the MPL by breaking down the problem into two phases.

In the first phase, the structural consistency constraint, corresponding to the restriction in problem (10), is omitted resulting in a relaxed problem consisting of d independent Markov blanket discovery problems:

$$\underset{mb(j)\subseteq V\setminus j}{\arg\max} \left[\log p(\mathbf{x}_j \mid \mathbf{x}_{mb(j)}) + \log p(mb(j))\right] \text{ for } j = 1, \ldots, d. \tag{11}$$

The first phase of the algorithm is thus similar in spirit to regression-based techniques meaning that the independent subproblems can be solved in parallel considerably improving real time efficiency. We refer to this as local MPL optimization since each node is optimized locally and independently of the other nodes.

By solving the relaxed problem, we obtain a collection of Markov blankets which usually are inconsistent with an undirected graph. The go-to approach among regression-based methods is simply to post-process the solution using either an AND-criterion ($\wedge$),

$$E_{\wedge} = \{\{i, j\} \in \{V \times V\} : i \in mb(j) \wedge j \in mb(i)\},$$

or an OR-criterion ($\vee$),

$$E_{\vee} = \{\{i, j\} \in \{V \times V\} : i \in mb(j) \vee j \in mb(i)\}.$$

From an MPL optimization perspective, neither of the above solutions is satisfactory. We therefore apply a second optimization phase whose goal is to combine the inconsistent Markov blankets from the first phase into a coherent structure which is MPL-optimal on a reduced model space determined by the relaxed solution. More specifically, the edge set given by $E_{\vee}$ is considered to be the result of a prescan that identifies eligible edges. We then construct a reduced model space from $E_{\vee}$ according to

$$\mathcal{G}_{\vee} = \{G \in \mathcal{G} : E \subseteq E_{\vee}\}. \tag{12}$$

The aim of the second phase is to solve the original problem with respect to the reduced model space:

$$\underset{G\in\mathcal{G}_{\vee}}{\arg\max} \left[\log \hat{p}(\mathbf{x} \mid G) + \log p(G)\right]. \tag{13}$$

The reduced model space $\mathcal{G}_{\vee}$ is in general considerably smaller than $\mathcal{G}$.

To summarize, instead of tackling the original problem (10), we propose the following approach:

1. Phase 1: Solve the relaxed problem (11).
2. Intermediate step: Form the reduced model space (12).
3. Phase 2: Solve the original problem on the reduced model space (13).

Under the proposed search algorithm, true edges can obviously be discarded during the first phase of the search. However, since the first phase attempts to maximize the MPL on a local level, the idea is that an edge that is not included from either direction in the first phase is less likely to be included in the second phase when attempting to maximize the MPL on a global level. Next we look at two simple approximate algorithms for solving the local and global problem in Phase 1 and Phase 2, respectively.

Phase 1 – An algorithm for local MPL optimization

To solve the relaxed problem, we basically need a Markov blanket discovery algorithm whose goal is to optimize the local MPL for each node independently of the solutions of the other nodes. For this we use an approximate deterministic hill climbing procedure similar to the interIAMB algorithm by Tsamardinos et al. (2003).

Pseudocode of the algorithm is presented in Algorithm 1 in Section S2 (Appendix) and the general idea is as follows. The algorithm is based on the two basic operations by which members are added to or deleted from the Markov blanket. The method is initiated with the empty Markov blanket and all other nodes are considered potential Markov blanket members. At each iteration, the method adds to the Markov blanket the node that induces the greatest improvement to the local MPL and updates the set of potential members accordingly. The algorithm interleaves each successful addition-step with a deletion phase. In the deletion phase, the algorithm removes the node that induces the largest improvement to score. The deletion-step is repeated until removal of a node no longer increases the score. When the addition-phase is iterated through without a successful addition, a local maximum has been reached, the algorithm terminates and returns the identified Markov blanket.

Phase 2 – An algorithm for global MPL optimization

As mentioned earlier, the variable-wise factorization of the MPL makes it particularly well-suited for search algorithms based on local edge changes. We therefore propose an algorithm that moves between neighboring graph structures. The set of neighbors of a graph G in a graph space $\mathcal{G}$ is denoted by $\mathcal{N}_{\mathcal{G}}(G)$ and defined as all graphs in $\mathcal{G}$ that can be reached from G by adding or removing a single edge.

Pseudocode of the algorithm is presented in Algorithm 2 in Section S2 (Appendix) and the general idea is as follows. The empty graph is set as the initial graph. At each iteration, all neighbors of the current graph in the considered graph space (in our case $\mathcal{G}_{\vee}$) are evaluated. At the end of the iteration, we choose the highest scoring graph from the neighbors, assuming that it has a higher score than the current graph, and repeat the procedure. If no candidate among the neighbors has a higher score than the current graph, a local maximum has been reached, the algorithm terminates and returns the identified graph.

By implementing smart caching, the efficiency of the algorithm can be improved considerably. First of all, by storing the log-scores of each node of the current graph, the only log-scores that need to calculated when evaluating a neighbor are those of the two nodes whose Markov blankets differ with respect to the current graph, that is, the two first terms in (9), given that G_2 is the current graph, along with the two corresponding prior terms. Moreover, most of the log-ratios (8) between the current graph and its neighbors will be preserved for the next graph and its corresponding neighbors. By storing the log-ratio of each neighbor from the previous iteration, only a small fraction of the neighbor set needs to be re-evaluated at each iteration. For example, for the complete graph space $\mathcal{G}$ over d nodes, only $2(d-1)$ of the $\binom{d}{2} - 1$ neighbors need to be re-evaluated.

5 Experimental results

The purpose of this section is to empirically investigate the performance of the MPL using the optimization algorithm described in the previous section. We evaluate our approach by comparing it to other state-of-the-art methods on synthetic models as well as known real-world networks. Since the graph structures of the generating models are known, it allows for a controlled and systematical comparison of the methods.

We assess the quality of an identified network structure by the Hamming distance which is the number of occurrences where an actual edge is missing or a non-edge is present. Consequently, a low value on the Hamming distance corresponds to structural resemblance to the true network structure and the minimum value of zero is obtained for the correct graph. In addition to structural resemblance, we monitor the execution times for the different methods. The methods in the experiments were implemented and run in MATLAB® (R2014a). For a more detailed overview of the results, see Section S3 in the Appendix.

5.1 Synthetic Markov networks

In the first experiment, we used synthetic models to generate datasets of different sizes. For simplicity, we restricted the models to binary variables. The synthetic graphs were formed by combining disconnected 16-node components representing various structural characteristics (see Figure 1). Initially, one replica of each graph component was combined to form a graph over 64 nodes. This procedure was then repeated with 2, 4, and 8 replicas to form graphs over 128, 256, and 512 nodes, respectively. Each final graph thereby contained all the structural characteristics present in the graph components. The disconnected nature of the final network facilitates the sampling procedure substantially since each disconnected component can be sampled independently of the rest of the network directly from its joint distribution. In practice, a distribution was generated by randomly sampling the maximal clique factors in (1). Each factor value $\phi(x_C)$ was drawn, independently of the other values, from a uniform distribution over $(0, 1)$. Consequently, the strength of the dependencies entailed by the edges may have varied considerably. To increase the stability of our results, for each sample and graph size, we generated 100 distributions and corresponding samples over which the final results were averaged. The experiments were performed for sample sizes ranging from 250 to 8000.

We compared the MPL against the following structure learning methods which are also applicable in high dimensions:

- **PIC:** The PIC criterion by Csiszár and Talata (2006) was applied using the exact same search technique as was introduced for the MPL method. From the proof of Theorem 1, we know that MPL and PIC result in asymptotically equivalent estimators, but here we examine how they perform in practice for limited sample sizes.
- **$\mathbf{PC_{skeleton}}$:** We applied the PC algorithm (Spirtes et al., 2000) which was developed for the purpose of identifying a partially directed acyclic graph (PDAG). Although the PC algorithm was originally intended for structure learning of Bayesian

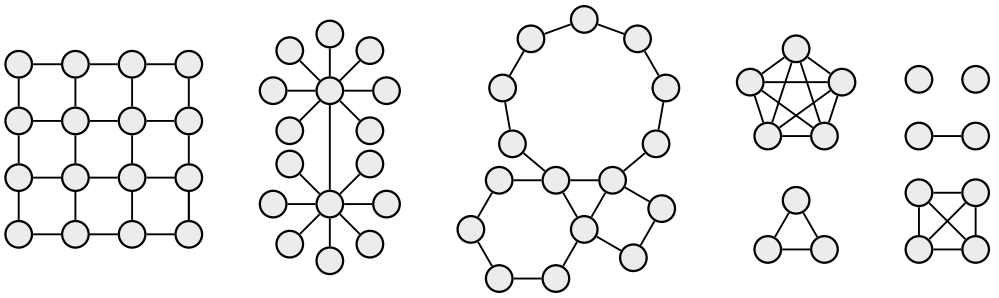

Figure 1: Four different 16-node graph components used to form the synthetic network structures.

networks, its initial phase is also directly applicable to Markov networks. Starting from a complete graph, the first phase of the method performs a series of independence tests to discover a so-called skeleton which captures all direct dependencies in form of an undirected graph. If the underlying model is a Markov network with a faithful distribution, the skeleton corresponds to the structure of the Markov network. To test for conditional independence we used a Bayesian score test which has been shown to perform better than a traditional χ^2-test in this context (Abellán et al., 2006). We set the hyperparameters to 0.5 and restricted the fan-in of the PC algorithm to 5, meaning that the maximum number of conditioning variables in a test was restricted to 5. The PC algorithm was performed using the Bayes net toolbox for MATLAB (Murphy, 2001).

- **L1LR-BIC$_\gamma$:** We applied the ℓ_1-regularized logistic regression (L1LR) approach by Barber and Drton (2015) which is directly applicable to our models since we have restricted the experiments to binary variables. The method is based on the approach by Ravikumar et al. (2010) and uses an extended Bayesian information criterion (BIC$_\gamma$) to select a Markov blanket among a set of candidates obtained from the regularization path. We set the value of the γ-parameter to 0.5. Since the collection of Markov blankets may contain inconsistencies, the final graph was formed using both the $\wedge$- and the $\vee$-criterion. The ℓ_1-regularized logistic regression was performed using the L1General package for MATLAB (Schmidt, 2010).

In Table 1 the average Hamming distances are listed for the different methods and model sizes. For details regarding true and false positives, see Table 1 in Section S3. Throughout the experiment, the MPL method obtained significantly lower Hamming distances than the other methods, which were overall quite equal. The PIC method was the most conservative of the methods resulting in an improved performance as the model size grew. The PC method, on the other hand, required a large sample size to keep down the number of false positives, especially when the model size was large. For the L1LR method, the $\vee$-version was slightly better than the $\wedge$-version as long as the sample size was large in relation to the model size, otherwise the $\wedge$-version was more stable. In terms of speed (see Table 2 in Section S3), the PC method was the fastest, however, the MPL- and PIC-based methods also performed at a comparable level. The

d	$n/1000$	MPL	PIC	$PC_{skeleton}$	L1LR-$BIC_{0.5}$ $\wedge$	L1LR-$BIC_{0.5}$ $\vee$
64	.25	50.0 ± 5.8	56.7 ± 3.5	56.8 ± 6.2	54.3 ± 4.5	53.4 ± 5.7
	.5	37.7 ± 6.1	48.9 ± 3.4	44.1 ± 5.6	44.6 ± 4.9	43.5 ± 5.0
	1	25.7 ± 4.7	39.1 ± 4.7	34.1 ± 5.2	35.3 ± 5.6	33.3 ± 5.8
	2	19.8 ± 3.9	30.4 ± 4.3	27.4 ± 4.1	28.2 ± 5.0	26.5 ± 4.9
	4	15.8 ± 3.5	23.3 ± 3.6	22.5 ± 3.9	21.9 ± 4.8	20.9 ± 4.7
	8	11.8 ± 2.7	18.4 ± 3.1	17.1 ± 2.9	15.5 ± 3.6	15.3 ± 4.2
128	.25	104.6 ± 8.4	114.0 ± 5.3	124.3 ± 9.8	111.3 ± 6.4	112.5 ± 8.4
	.5	75.6 ± 8.0	97.0 ± 5.4	93.4 ± 7.7	90.8 ± 6.8	89.0 ± 7.6
	1	55.1 ± 6.7	79.4 ± 5.9	73.9 ± 7.0	74.8 ± 7.1	72.3 ± 7.9
	2	39.5 ± 4.9	59.8 ± 5.6	56.2 ± 5.7	56.8 ± 6.3	55.0 ± 6.2
	4	31.8 ± 5.1	46.3 ± 5.2	45.1 ± 5.8	44.6 ± 7.5	42.3 ± 7.4
	8	24.7 ± 4.7	36.4 ± 4.5	35.1 ± 5.1	33.6 ± 6.3	32.8 ± 6.7
256	.25	218.8 ± 11.8	230.2 ± 7.5	278.4 ± 12.6	228.4 ± 8.5	240.4 ± 10.5
	.5	164.1 ± 12.3	196.6 ± 8.7	207.2 ± 13.5	191.6 ± 9.6	195.1 ± 11.5
	1	113.2 ± 9.9	159.1 ± 8.3	155.5 ± 9.9	152.3 ± 9.2	152.5 ± 10.0
	2	82.8 ± 7.9	119.6 ± 8.3	119.1 ± 9.6	120.6 ± 9.6	118.2 ± 11.3
	4	63.5 ± 6.7	91.1 ± 6.9	91.9 ± 7.7	91.5 ± 9.9	89.8 ± 9.6
	8	50.1 ± 6.0	73.1 ± 5.4	72.1 ± 7.1	69.5 ± 8.3	69.0 ± 11.2
512	.25	464.4 ± 17.7	475.1 ± 13.1	632.2 ± 22.5	473.6 ± 13.6	519.2 ± 16.3
	.5	342.5 ± 18.7	396.8 ± 10.7	460.3 ± 18.4	395.5 ± 12.6	416.6 ± 18.2
	1	236.7 ± 13.8	318.2 ± 11.1	335.1 ± 15.6	317.5 ± 12.6	326.6 ± 16.2
	2	170.0 ± 11.9	240.9 ± 12.7	251.2 ± 14.4	249.8 ± 15.4	253.5 ± 16.5
	4	130.9 ± 9.7	184.0 ± 9.4	193.1 ± 12.3	192.2 ± 15.4	194.0 ± 15.5
	8	103.0 ± 8.7	149.1 ± 8.8	149.9 ± 9.7	145.5 ± 12.7	146.7 ± 13.8

Table 1: Hamming distances (mean $\pm$ standard deviation) in the synthetic network experiments (d = number of nodes, n = sample size). The shade of the cell colors represents the ranking of the methods for each row such that the lightest color marks the lowest Hamming distance and the darkest color marks the highest Hamming distance.

L1LR method was the slowest and most negatively affected by an increased model size. Note, however, that both the L1LR optimization and the first phase of the MPL/PIC search (which is the most time-consuming) can be executed in a completely parallel fashion.

5.2 Real-world Bayesian networks

In the second experiment, we performed experiments on well-known real-world models, from the related class of Bayesian networks, in a similar fashion as Bromberg et al. (2009). The considered models are commonly used as benchmarks in research and are available from a number of sources. The networks used in this work were obtained from the Bayesian network repository at http://www.bnlearn.com/bnrepository/ and sampled using the R package by Scutari (2010).

Network	Alarm	Andes	Barley	Hailfinder	Insurance	Win95pts
Number of nodes	37	223	48	56	27	76
Number of edges (moral graph)	65	626	126	99	70	225
Number of parameters	509	1157	114005	2656	984	574
Markov blanket size	1–8	0–23	2–13	1–17	1–10	1–29
Variable cardinality	2–4	2	2–67	2–11	2–5	2

Table 2: Properties of the real-world Bayesian networks.

To transform the directed acyclic graph of a Bayesian network into a corresponding undirected graph of a Markov network, a two-step procedure known as moralization is used (see Lauritzen, 1996; Koller and Friedman, 2009). In the first step, all parents of a common child are connected by an undirected edge if not already connected. In the second step the graph is made undirected by removing the direction of all directed edges. Although the local Markov property remains valid in the transformed network, some conditional independencies are lost in the moralization process due to the added edges. Consequently, the associated distribution is no longer faithful to the undirected graph making the graph identification more challenging.

We selected six networks which are listed along with some of their properties in Table 2. Compared to the relatively simple and balanced synthetic networks in Section 5.1, these models are more challenging due to their higher edge density and larger Markov blankets. In addition, there are now also non-binary variables. As before, we sampled each network for sample sizes ranging from 250 to 8000. For each network and sample size, we generated 100 samples over which the final results were averaged. We applied the same methods as in the previous section under some modifications. Since the true model is now a Bayesian network, for which the PC algorithm is designed, we now also apply the second phase which directs some of the edges resulting in a PDAG. The final undirected graph was obtained by moralizing the PDAG. Furthermore, since the L1LR under the current implementation is restricted to binary variables, we only applied it to the two networks which consisted solely of binary variables; *Andes* and *Win95pts*.

In Table 3 the average Hamming distances from the moralized graph are listed for the different methods and networks. For details regarding true and false positives, see Table 3 in Section S3. Again, the MPL performed very well and almost consistently achieved lower distances than the other methods. The PIC method was again very conservative with few false positives. The PC algorithm performed quite well under this setup, which could be expected considering that the true models were Bayesian networks. An interesting observation regarding the L1LR method is that number of false positives increased as the sample size was increased. A possible reason for this is that the current L1LR method only includes pairwise interactions, by which it attempts to approximate all higher-order interactions. In contrast to the synthetic models, which mainly contained pairwise interactions and for which the number of false positives steadily decreased, the Bayesian networks contained a lot of interactions of higher order than two. In terms of speed (see Table 4 in Section S3), the results followed the same pattern as in the previous experiment.

Network	$n/_{1000}$	MPL	PIC	PC_{moral}	L1LR-$BIC_{0.5}$ $\wedge$	L1LR-$BIC_{0.5}$ $\vee$
Alarm	.25	31.7 ± 2.6	49.2 ± 1.4	34.3 ± 3.1	–	–
	.5	24.7 ± 1.6	47.2 ± 1.0	29.4 ± 3.0	–	–
	1	20.0 ± 1.5	41.5 ± 1.3	25.1 ± 2.8	–	–
	2	16.8 ± 0.7	34.5 ± 1.7	21.0 ± 2.6	–	–
	4	15.4 ± 0.8	31.1 ± 0.8	18.7 ± 1.9	–	–
	8	14.0 ± 0.5	26.3 ± 1.2	17.6 ± 1.8	–	–
Andes	.25	412.0 ± 7.3	488.6 ± 4.4	491.2 ± 15.4	494.9 ± 6.5	482.3 ± 8.9
	.5	362.5 ± 5.7	451.8 ± 3.4	419.5 ± 10.8	445.1 ± 5.3	437.6 ± 7.6
	1	327.8 ± 5.3	422.9 ± 3.3	363.9 ± 10.1	402.7 ± 4.8	397.2 ± 9.6
	2	295.9 ± 4.7	390.7 ± 2.9	319.3 ± 8.3	363.2 ± 5.7	360.1 ± 10.5
	4	258.5 ± 6.3	357.8 ± 2.6	271.3 ± 7.0	329.0 ± 4.1	324.9 ± 9.1
	8	216.7 ± 4.6	331.4 ± 4.1	233.4 ± 7.9	311.0 ± 3.8	297.3 ± 6.5
Barley	.25	105.0 ± 1.1	115.0 ± 2.2	113.8 ± 2.9	–	–
	.5	100.6 ± 1.4	108.3 ± 1.0	112.6 ± 2.1	–	–
	1	95.8 ± 1.3	104.2 ± 0.9	108.7 ± 2.0	–	–
	2	93.8 ± 1.8	100.2 ± 0.4	102.7 ± 2.4	–	–
	4	88.3 ± 1.0	98.4 ± 0.5	94.2 ± 2.3	–	–
	8	86.7 ± 0.9	94.4 ± 1.1	91.1 ± 2.0	–	–
Hailfinder	.25	69.8 ± 1.6	81.1 ± 1.4	76.9 ± 3.5	–	–
	.5	67.0 ± 1.3	76.6 ± 1.3	73.6 ± 3.3	–	–
	1	64.3 ± 1.6	70.3 ± 1.1	73.8 ± 3.6	–	–
	2	59.3 ± 3.0	67.2 ± 1.4	72.9 ± 3.0	–	–
	4	55.9 ± 1.8	64.5 ± 2.4	74.0 ± 3.3	–	–
	8	50.6 ± 2.9	59.3 ± 0.8	72.3 ± 3.8	–	–
Insurance	.25	50.2 ± 2.2	56.3 ± 1.4	48.9 ± 2.7	–	–
	.5	45.0 ± 1.7	54.5 ± 1.3	45.2 ± 2.9	–	–
	1	40.2 ± 1.6	50.8 ± 1.8	40.2 ± 1.9	–	–
	2	37.5 ± 1.0	49.1 ± 0.9	38.3 ± 1.8	–	–
	4	34.9 ± 1.4	45.0 ± 0.9	36.1 ± 2.7	–	–
	8	31.8 ± 1.1	42.2 ± 1.0	33.6 ± 1.0	–	–
Win95pts	.25	172.0 ± 5.6	192.3 ± 2.4	193.5 ± 7.1	196.2 ± 3.3	180.7 ± 5.5
	.5	147.3 ± 6.0	183.9 ± 2.6	180.4 ± 6.6	187.1 ± 3.6	168.2 ± 6.2
	1	118.6 ± 5.5	173.8 ± 2.5	163.1 ± 7.4	176.4 ± 3.6	152.6 ± 6.4
	2	99.8 ± 3.0	157.3 ± 2.1	143.9 ± 6.5	162.8 ± 3.9	141.6 ± 6.3
	4	89.5 ± 2.6	144.2 ± 1.5	126.9 ± 6.0	149.0 ± 3.5	135.1 ± 7.0
	8	80.1 ± 2.2	132.5 ± 1.6	109.9 ± 6.4	136.6 ± 3.6	132.3 ± 7.2

Table 3: Hamming distances (mean $\pm$ standard deviation) in the Bayesian network experiments (n = sample size). The shade of the cell colors represents the ranking of the methods for each row such that the lightest color marks the lowest Hamming distance and the darkest color marks the highest Hamming distance.

6 Conclusions

In this work we have introduced a novel approach for learning the graph structure of a Markov network without imposing the restriction of chordality. Our MPL score is proven to be consistent and can be considered a small sample analytical version of the information theoretic PIC criterion (Csiszár and Talata, 2006). Furthermore, we have discussed the connection between the MPL for Markov networks and what would be the marginal likelihood of bi-directional dependency networks under a complete parameterization.

Since the MPL-based optimization problem is intractable, we designed an efficient search algorithm that exploits the decomposable structure of the MPL. Under the proposed optimization strategy, our MPL method is similar in spirit to the max–min hill climbing algorithm for learning Bayesian networks (Tsamardinos et al., 2006). The main difference is that both phases of our algorithm are derived from the notion of maximizing a single underlying score.

In our experiments on both synthetic and real-world networks, our MPL method outperformed its competitors in terms of Hamming distance between the identified and true graph. In addition, the execution times demonstrate the applicability of our method on high-dimensional systems, in particular, when considering the possibility of parallelizing the first phase of the search method.

There has been a lot of research in finding Bayesian networks that maximize the Bayesian score. In particular, there has recently been a growing interest of using computational logic for structure learning (Bartlett and Cussens, 2013; Corander et al., 2013; Berg et al., 2014; Parviainen et al., 2014). In the future it would be interesting to develop a comparable approach for exact global optimization of the MPL score under meaningful restrictions to retain computational scalability.

The main drawback of the MPL is that it, as a result of the parameter independence assumption (6), in a sense over-specifies the node-wise conditional distributions. This has a negative effect on the data-efficiency of the MPL, especially for hub nodes, since the conditional distributions are specified in terms of complete Markov blankets even if only a subset of a Markov blanket is sufficient for shielding a node from a particular part of the network. Taking this observation into account in future research, we will look into improving the data-efficiency of the MPL. Additional directions for future research are extending the scope of the MPL by considering continuous variables as well as combining the MPL with the models of Nyman et al. (2014) and Pensar et al. (2015) in order to enable efficient learning of non-chordal context-specific Markov network structures.

Supplementary Material

Supplementary Appendix to "Marginal Pseudo-Likelihood Learning of Discrete Markov Network Structures" (DOI: 10.1214/16-BA1032SUPP; .pdf). The appendix contains a proof of the consistency theorem, pseudocode of the search algorithms, and detailed results from the numerical experiments.

References

Abellán, J., Gómez-Olmedo, M., and Moral, S. (2006). "Some variations on the PC algorithm." In *Proceedings of the 3rd European Workshop on Probabilistic Graphical Models*, 1–8. 1208

Akaike, H. (1974). "A new look at the statistical model identification." *IEEE Transactions on Automatic Control*, 19: 716–723. MR0423716. 1199

Anandkumar, A., Tan, V. Y. F., Huang, F., and Willsky, A. S. (2012). "High-dimensional structure estimation in Ising models: Local separation criterion." *The Annals of Statistics*, 40: 1346–1375. MR3015028. doi: http://dx.doi.org/10.1214/12-AOS1009. 1198

Aurell, E. and Ekeberg, M. (2012). "Inverse Ising inference using all the data." *Physical Review Letters*, 108: 090201. 1196

Barber, R. F. and Drton, M. (2015). "High-dimensional Ising model selection with Bayesian information criteria." *Electronic Journal of Statistics*, 9(1): 567–607. MR3326135. doi: http://dx.doi.org/10.1214/15-EJS1012. 1196, 1202, 1203, 1208

Bartlett, M. and Cussens, J. (2013). "Advances in Bayesian Network Learning using Integer Programming." In *Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence*, 182–191. 1212

Berg, J., Järvisalo, M., and Malone, B. (2014). "Learning optimal bounded treewidth Bayesian networks via maximum satisfiability." In *Proceedings of the 17th Conference on Artificial Intelligence and Statistics*, 86–95. 1212

Besag, J. (1975). "Statistical analysis of non-lattice data." *Journal of the Royal Statistical Society. Series D (The Statistician)*, 24: 179–195. 1196, 1199

Bromberg, F., Margaritis, D., and Honavar, V. (2009). "Efficient Markov network structure discovery using independence tests." *Journal of Artificial Intelligence Research*, 35: 449–485. MR2534496. 1198, 1209

Chow, C. and Liu, C. (1968). "Approximating discrete probability distributions with dependence trees." *IEEE Transactions on Information Theory*, 14(3): 462–467. 1199

Corander, J., Janhunen, T., Rintanen, J., Nyman, H., and Pensar, J. (2013). "Learning chordal Markov networks by constraint satisfaction." In *Advances in Neural Information Processing Systems 26*, 1349–1357. 1212

Csiszár, I. and Talata, Z. (2006). "Consistent estimation of the basic neighborhood of Markov random fields." *Annals of Statistics*, 34: 123–145. MR2275237. doi: http://dx.doi.org/10.1214/009053605000000912. 1196, 1200, 1207, 1212

Ekeberg, M., Lövkvist, C., Lan, Y., Weigt, M., and Aurell, E. (2013). "Improved contact prediction in proteins: Using pseudolikelihoods to infer Potts models." *Physical Review E*, 87: 012707. 1196

Heckerman, D., Chickering, D. M., Meek, C., Rounthwaite, R., and Kadie, C. (2000).

"Dependency networks for inference, collaborative filtering, and data visualization." *Journal of Machine Learning Research*, 1: 49–75. 1196, 1201, 1203

Heckerman, D., Geiger, D., and Chickering, D. M. (1995). "Learning Bayesian networks: The combination of knowledge and statistical data." *Machine Learning*, 20: 197–243. 1196, 1200, 1201

Höfling, H. and Tibshirani, R. (2009). "Estimation of sparse binary pairwise Markov networks using pseudo-likelihoods." *Journal of Machine Learning Research*, 10: 883–906. MR2505138. 1196, 1198, 1200

Ji, C. and Seymour, L. (1996). "A consistent model selection procedure for Markov random fields based on penalized pseudolikelihood." *Annals of Applied Probability*, 6: 423–443. MR1398052. doi: http://dx.doi.org/10.1214/aoap/1034968138. 1196, 1200

Koller, D. and Friedman, N. (2009). *Probabilistic Graphical Models: Principles and Techniques*. MIT Press. MR2778120. 1195, 1198, 1200, 1210

Lauritzen, S. L. (1996). *Graphical Models*. Oxford: Oxford University Press. MR1419991. 1195, 1210

Lee, S.-I., Ganapathi, V., and Koller, D. (2006). "Efficient structure learning of Markov networks using ℓ_1-regularization." In *Advances in Neural Information Processing Systems 19*, 817–824. 1198, 1199, 1200

Liu, Q. and Ihler, A. T. (2012). "Distributed parameter estimation via pseudo-likelihood." In *Proceedings of the 29th International Conference on Machine Learning*, 1487–1494. 1198

Lowd, D. and Davis, J. (2014). "Improving Markov network structure learning using decision trees." *Journal of Machine Learning Research*, 15: 501–532. 1196, 1198, 1203

Meinshausen, N. and Bühlmann, P. (2006). "High-dimensional graphs and variable selection with the lasso." *The Annals of Statistics*, 34(3): 1436–1462. MR2278363. doi: http://dx.doi.org/10.1214/009053606000000281. 1196, 1203

Mizrahi, Y. D., Denil, M., and de Freitas, N. (2014). "Linear and parallel learning of Markov random fields." In *Proceedings of the 31st International Conference on Machine Learning*, 199–207. 1198

Murphy, K. P. (2001). "The Bayes net toolbox for MATLAB." *Computing Science and Statistics*, 33: 1024–1034. 1208

Nyman, H., Pensar, J., Koski, T., and Corander, J. (2014). "Stratified graphical models – context-specific independence in graphical models." *Bayesian Analysis*, 9(4): 883–908. MR3293960. doi: http://dx.doi.org/10.1214/14-BA882. 1212

Parviainen, P., Farahani, H., and Lagergren, J. (2014). "Learning bounded tree-width Bayesian networks using integer linear programming." In *Proceedings of the 17th Conference on Artificial Intelligence and Statistics*, 751–759. 1212

Pensar, J., Nyman, H., Niiranen, J., and Corander, J. (2016). "Supplementary appendix to "Marginal pseudo-likelihood learning of discrete Markov network structures"." *Bayesian Analysis*. doi: http://dx.doi.org/10.1214/16-BA1032SUPP. 1197

Pensar, J., Nyman, H., Koski, T., and Corander, J. (2015). "Labeled directed acyclic graphs: A generalization of context-specific independence in directed graphical models." *Data Mining and Knowledge Discovery*, 29(2): 503–533. MR3312469. doi: http://dx.doi.org/10.1007/s10618-014-0355-0. 1212

Pietra, S. D., Pietra, V. D., and Lafferty, J. (1997). "Inducing features of random fields." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19: 380–393. 1198

Ravikumar, P., Wainwright, M. J., and Lafferty, J. D. (2010). "High-dimensional Ising model selection using ℓ_1-regularized logistic regression." *Annals of Statistics*, 38: 1287–1319. MR2662343. doi: http://dx.doi.org/10.1214/09-AOS691. 1196, 1198, 1200, 1203, 1208

Schmidt, M. (2010). "L1General." https://www.cs.ubc.ca/~schmidtm/Software/L1General.html. 1208

Schwarz, G. (1978). "Estimating the dimension of a model." *Annals of Statistics*, 6: 461–464. MR0468014. 1199, 1200

Scutari, M. (2010). "Learning Bayesian networks with the bnlearn R package." *Journal of Statistical Software*, 35(3): 1–22. 1209

Spirtes, P., Glymour, C., and Scheines, R. (2000). *Causation, Prediction, and Search*. MIT Press, 2nd edition. MR1815675. 1198, 1207

Tsamardinos, I., Aliferis, C., Statnikov, A., and Statnikov, E. (2003). "Algorithms for large scale Markov blanket discovery." In *The 16th International FLAIRS Conference*, 376–380. 1198, 1206

Tsamardinos, I., Brown, L. E., and Aliferis, C. F. (2006). "The max–min hill-climbing Bayesian network structure learning algorithm." *Machine Learning*, 65: 31–78. 1212

Whittaker, J. (1990). *Graphical Models in Applied Multivariate Statistics*. Chichester: Wiley. MR1112133. 1198

Acknowledgments

JP was supported by the Magnus Ehrnrooth Foundation. HN was supported by the Foundation of Åbo Akademi University, as part of the grant for the Center of Excellence in Optimization and Systems Engineering. JC was supported by the Academy of Finland grant 251170. The authors would like to thank the AE and the two anonymous reviewers for their comments and suggestions which led to a significant improvement of the original manuscript.